

Proof assistants for teaching: a survey

Frédéric Tran-Minh¹ Laure Gonnord¹ Julien Narboux²

¹LCIS - Grenoble INP - UGA

²ICube - University of Strasbourg

ThEdu24, Nancy, France - July 1st, 2024



ThEdu'24

Motivations

Starting point

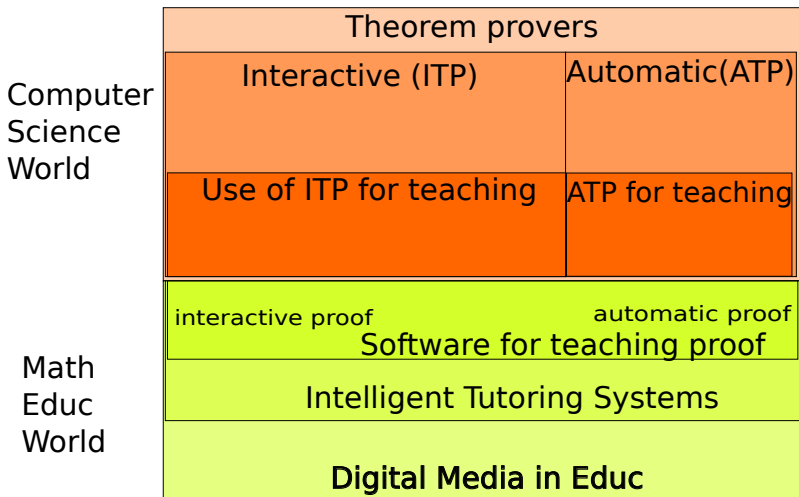
phD thesis about the use of proof assistants for teaching

- As far as we know : no survey ?

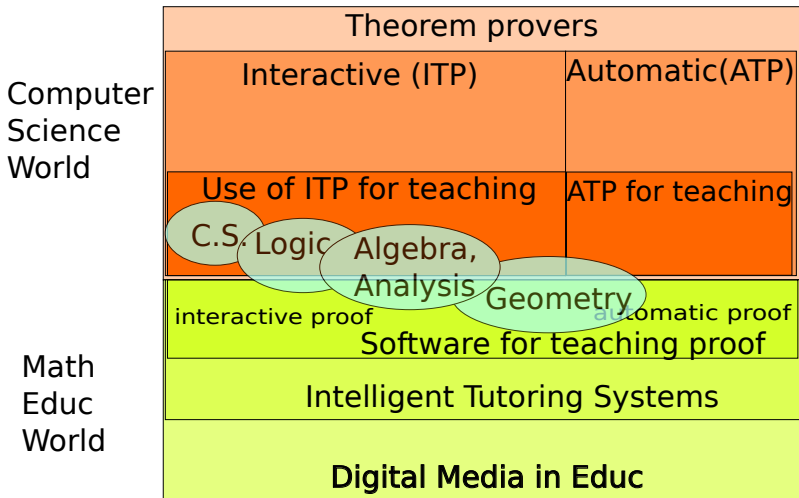
Focus

	Theorem provers	
Computer Science World	Interactive (ITP)	Automatic(ATP)
	Use of ITP for teaching	ATP for teaching

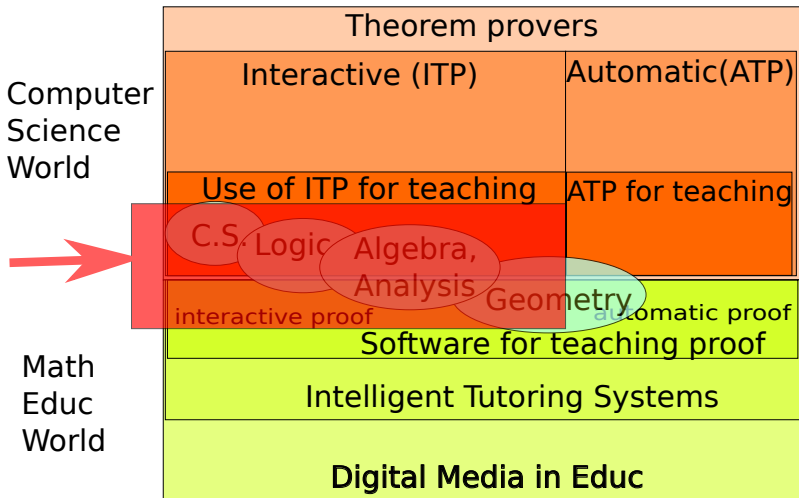
Focus



Focus



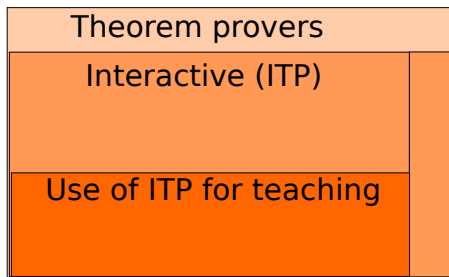
Focus



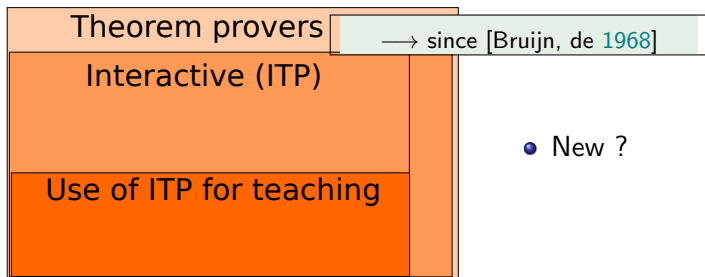
Contents

- 1 Reports on teaching experiments
- 2 Different categories of proof assistants for teaching
- 3 Input language
- 4 Feedback
- 5 Underlying theory
- 6 Teaching environment
- 7 Conclusion
- 8 Bibliography

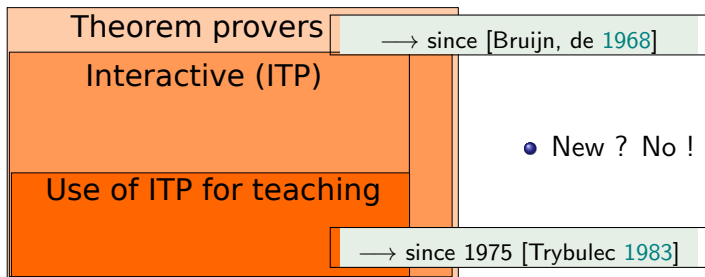
Reports of teaching experiments



Reports of teaching experiments



Reports of teaching experiments



Reports of teaching experiments

Theorem provers

→ since [Bruijn, de 1968]

Interactive (ITP)

• New ? No !

Use of ITP for teaching

C.S. Logic

Algebra, Analysis

→ since 1975 [Trybulec 1983]

Geometry

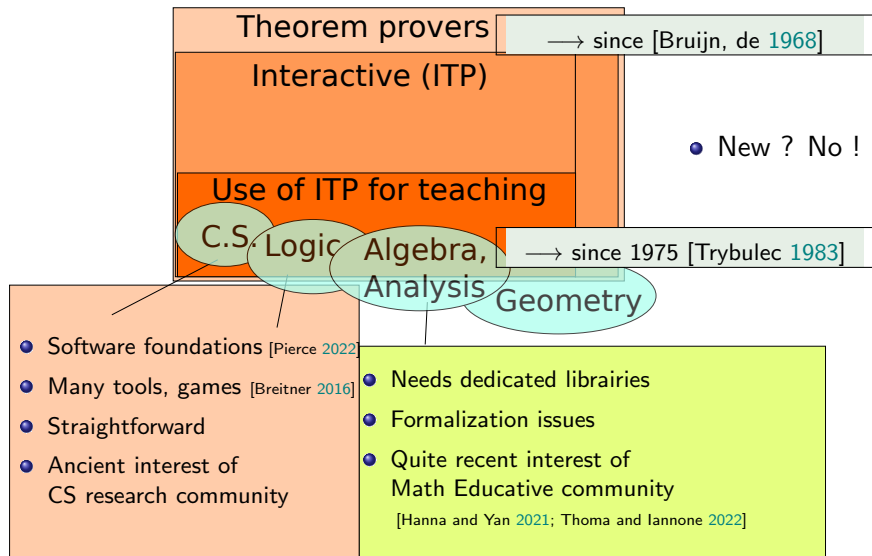
[Pierce 2022; Delahaye, Jaume, and Prevosto 2005; Bertot 2015 ...]

[Retel and Zalewska 2005; C. K. F. Wiedijk 2007; The Cocorico Coq wiki 2021; From, Villadsen, and Blackburn 2020 ...]

[Blanc et al. 2007; Kerjean et al. 2022; Macbeth 2023 ...]

[Richard and Fortuny 2007; Webber et al. 2001 ...]

Reports of teaching experiments



Different categories of proof assistants for teaching

- Dedicated proof assistants : teaching community (Didactics, ITS)
 - For logic and undergraduate maths : *CPT, CalcCheck, Tutch, XBarnacle/CLAM, WinKE, ProveEasy, Jape, ProofLab, Omega Tutor, PhoX, TPE (Epgy), EasyProve, Lurch, Diproche, Edukera*
 - For Geometry: *AgentGeom, Baghera, Chypre, Cabri Euclide, Geometrix, Geometry Tutor, Geometry Explanation Tutor, Angle, Mentonizeh, QED-Tutrix, Turing, Advanced Geometry Tutor, Geometrix*
- General purpose proof assistants : CS community
Mizar, Coq, Isabelle/Isar, Lean, HOL-Light, ...
- Adaptations of general purpose proof assistants for general use
 - On top of Coq : *CoqIDE, jsCoq, ProofGeneral*
 - On top of Isabelle : *XIsabelle, Isabelle/jEdit, ProofGeneral*
 - On top of Lean : *Lean Web Editor, ProofWidgets, Paperproof*
- Adaptations of general purpose proof assistants for teaching
 - On top of Coq : *PCoq, Papuq, CoqWeb, Waterproof, ProofWeb, TryLogic, Geoview, GeoProof*
 - On top of Lean : *Deaduction, Lean Verbose*
 - Other : *ETPS on top of TPS*

Input language

(The Automath language) is difficult to write and to read. It's true, but - in the sense of programming languages - we can compare it to machine language, and we hope that we will manage to build simpler languages, that can be written and read more easily, and that can be translated by a computer in our "machine language".

De Bruijn 1969

Input language

- Bare proof terms

4. Chaque suite constante est convergente.

4.11	n	aa	:=	w(p,c,delta,n ₀ ,n)	bool
4.12	n	b	:=	then 2([u,TRUE(a)] abbrev 2,[u,TRUE(a)]then)	TRUE(aa)
4.13	n ₀	h	:=	then 2([s,nat] TRUE(aa(s)), [s,nat] b(s))	TRUE(z(p,c,delta,n ₀))
4.14	ass	d	:=	then 13*(nat,[x,nat]z(p,c,delta,1),1,h(1))	TRUE(y(p,c,delta))
4.15	delta	e	:=	then 2([x,abbrev 1]TRUE(y(p,c,delta)), [x,abbrev 1]d(x))	TRUE(q(p,c,delta))
4.16	c	f	:=	then 2([s,real]TRUE(q(p,c,s)),[s,real]e(s))	TRUE(lim(p,c))
4.17	c	g	:=	then 13*(real,[s,real]lim(p,s),c,f)	TRUE(conv(p))

Proof terms in Automath language (De Bruijn 1969).

Input language

- Bare proof terms

```

lemma and_iff_of_imp : ∀ P Q : Prop, (P → Q) → ((P ∧ Q) ↔ P) :=
  λ P Q : Prop ↪
    λ h_PimpQ : P → Q ↪
      Iff.intro
        (
          λ h_PandQ : P ∧ Q ↪
            (h_PandQ.left : P)
        )
        (
          λ h_P : P ↪
            have h_Q : Q := h_PimpQ h_P
            (And.intro (h_P : P) (h_Q : Q) : (P ∧ Q) )
        )
  )

```

Lean Proof term mode.

Input language

- Bare proof terms
- Core tactic-language

```

1 |Lemma and_iff_of_imp : ∀ P Q :  $\mathbb{P}$ , (P → Q) → ((P ∧ Q) ↔ P).
2 |Proof.
3 |  intros.
4 |  constructor.
5 |
6 |  intro H_PandQ.
7 |  destruct H_PandQ as [HP HQ].
8 |  exact HP.
9 |
10 |  intro HP.
11 |  constructor.
12 |  exact HP.
13 |  apply H.
14 |  exact HP.
15 |Qed.

```

Coq Procedural tactic language.

Input language

- Bare proof terms
- Core tactic-language
- Declarative style

```

50 Lemma "(P → Q) → ((P ∧ Q) ↔ P)"
51 proof
52   assume h: "P → Q"
53   have h1: "(P ∧ Q) → P"
54   proof
55     assume "P ∧ Q"
56     thus "P" ..
57   qed
58   have h2: "P → (P ∧ Q)"
59   proof
60     assume hP : P
61     with h have Q ..
62     with hP show "P ∧ Q" ..
63   qed
64   from h1 and h2 show "((P ∧ Q) ↔ P)" by blast
65 qed

```

- Mizar Matuszewski and Rudnicki 2005
- Isabelle/Isar Wenzel 2007


Isabelle/Isar declarative language.

Input language

- Bare proof terms
- Core tactic-language
- Declarative style
- Controlled Natural Languages (CNL)

Example Diproche Text 1

Es sei x eine ganze Zahl. Zeige: Wenn x gerade ist, dann ist $2-3*x$ gerade.

Beweis: Es sei x gerade. Dann gibt es eine ganze Zahl k mit $x=2*k$. Sei k eine ganze Zahl mit $x=2*k$. Dann ist $2-3*x=2-3*(2*k)=2*(1-3*k)$. Also ist $2-3*x$ gerade. qed 

“Let x be an integer. Prove: If x is even, then $2-3x$ is even.

Proof: Let x be even. Then there is an integer k such that $x = 2k$. Let k be an integer with $x = 2k$. Then we have $2-3x = 2-3 \cdot (2k) = 2(1-3k)$. Hence $2-3x$ is even. qed.

- SAD and ForTheL, Naproche [Lyaletski and Verchinine 2010 Kühlwein et al. 2009]
- CNL as tactic languages
 - PlatΩ / Ωmega [Wagner, Autexier, and Benz Müller 2007]
 - Diproche [Carl, Lorenzen, and Schmi]
 - Lean Verbose [Massot 2021]
 - Waterproof [Portegies et al. 2022]

Input language

- Bare proof terms
- Core tactic-language
- Declarative style
- Controlled Natural Languages (CNL)

Example "The squeeze theorem."

Given: $(u \ v \ w : \mathbb{N} \rightarrow \mathbb{R}) \ (l : \mathbb{R})$

Assume: $(hu : u \text{ converges to } l) \ (hw : w \text{ converges to } l)$
 $(h : \forall n, u \ n \leq v \ n) \ (h' : \forall n, v \ n \leq w \ n)$

Conclusion: v converges to l

Proof:

Fix $\varepsilon > 0$

Since u converges to l and $\varepsilon > 0$ we get N such that
 $hN : \forall n \geq N, |u \ n - l| \leq \varepsilon$

Since w converges to l and $\varepsilon > 0$ we get N' such that
 $hN' : \forall n \geq N', |w \ n - l| \leq \varepsilon$

Let's prove that $\max N \ N'$ works: $\forall n \geq \max N \ N', |v \ n - l| \leq \varepsilon$

Fix $n \geq \max N \ N'$

Since $n \geq \max N \ N'$ we get $(hn : n \geq N)$ and $(hn' : n \geq N')$
 Since $\forall n \geq N, |u \ n - l| \leq \varepsilon$ and $n \geq N$ we get
 $(hNl : -\varepsilon \leq u \ n - l)$ and $(hNd : u \ n - l \leq \varepsilon)$

Since $\forall n \geq N', |w \ n - l| \leq \varepsilon$ and $n \geq N'$ we get
 $(hN'l : -\varepsilon \leq w \ n - l)$ and $(hN'd : w \ n - l \leq \varepsilon)$

Let's prove that $|v \ n - l| \leq \varepsilon$

- SAD and ForTheL, Naproche [Lyaletski and Verchinine 2010 Kühlwein et al. 2009]

- CNL as tactic languages

- Plat Ω / Ω mega

[Wagner, Autexier, and Benz Müller 2007]

- Diproche [Carl, Lorenzen, and Schmitt 2007]
- Lean Verbose [Massot 2021]
- Waterproof [Portegies et al. 2022]

Input language

- Bare proof terms
- Core tactic-language
- Declarative style
- Controlled Natural Languages (CNL)

Lemma exercise_1 : 2 is the infimum of [2, 5].

Proof.

```

We need to show that
  (2 is a lower bound for [2, 5] ∧
   (for all m : ℝ, m is a lower bound for [2, 5] ⇒ m ≤ 2)).
We show both statements.
- We need to show that (2 is a lower bound for [2, 5]).
  We need to show that (for all x : ℝ, x : [2, 5] ⇒ 2 ≤ x).
  Take x : ℝ. Assume that (x : [2, 5]).
  We conclude that (2 ≤ x).
- We need to show that
  (for all m : ℝ, m is a lower bound for [2, 5] ⇒ m ≤ 2).
Take m : ℝ. Assume that (m is a lower bound for [2, 5]).
It holds that (2 : [2, 5]).
We conclude that (m ≤ 2).
  
```

Qed.

- SAD and ForTheL, Naproche [Lyaletski and Verchinine 2010 Kühlwein et al. 2009]

- CNL as tactic languages

- PlatΩ / Ωmega

[Wagner, Autexier, and Benz Müller 2007]

- Diproche [Carl, Lorenzen, and Schmi
- Lean Verbose [Massot 2021]
- Waterproof [Portegies et al. 2022]

Waterproof

Input language

- Bare proof terms
- Core tactic-language
- Declarative style
- Controlled Natural Languages (CNL)
- Point-and-click graphical input

- CtCoq / PCoq [Bertot 1999; Amerkad et al.]

The screenshot shows a Coq proof editor window titled "Command • State • Search Theorems <1>". The main area displays a theorem named "second_degree_decomposition" with the following statement:

$$\forall a, b, x, \text{discr} : \mathbb{R}, \text{discr} = a^2 - 4 \cdot b \Rightarrow (\text{discr} \geq 0) \Rightarrow x^2 + a \cdot x + b = (x + (a + \sqrt{|\text{discr}|}) / 2) \cdot (x + (a - \sqrt{|\text{discr}|}) / 2).$$

Below the theorem, the proof state is shown with the following elements:

- Goal: $x^2 + a \cdot x + b = (x + a / 2 + \sqrt{|\text{discr}|} / 2) \cdot (x + (a - \sqrt{|\text{discr}|}) / 2)$
- Hypothesis $H'0$: $(\text{discr} \geq 0)$
- Hypothesis H' : $\text{discr} = a^2 - 4 \cdot b$
- Context: $\text{discr} : \mathbb{R}$

Figure 3: Notations for real number algebraic calculus.

Input language

- Bare proof terms
- Core tactic-language
- Declarative style
- Controlled Natural Languages (CNL)
- Point-and-click graphical input

The screenshot shows the 'SampleRootTwo - Proof Ed' window. On the left, a list of definitions is visible, including 'Definition of divides relation', 'Definition of Even', 'Definition of irrational number', 'Definition of rational number', 'Definition of squaring function', and 'Property of even product'. The main proof window on the right shows a goal '(A.3) $2 = \frac{(m)^2}{(n)^2}$ ' and a 'Derivation System' dialog box with various icons for applying rules. Below the proof window, the text '(A.3) $\neg(\sqrt{2} \text{ is rational})$ By A' is visible.

- CtCoq / PCoq [Bertot 1999; Amerkad et al]
- Predicate Prover [Abrial and Cansell 2003]
- TPE / Epgy [Sommer and Nuckols 2004]

TPE / Epgy

Figure 5. Application of the Derivation System.

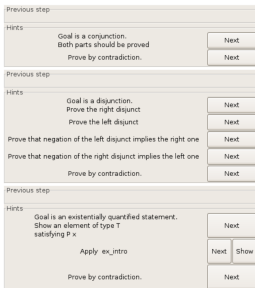
Input language

- Bare proof terms
- Core tactic-language
- Declarative style
- Controlled Natural Languages (CNL)
- Point-and-click graphical input

A : Prop
 B : Prop
 -----(1/1)
 A ∧ B

A : Prop
 B : Prop
 -----(1/1)
 A ∨ B

T : Type
 P : T → Prop
 -----(1/1)
 exists x:T, P x



- CtCoq / PCoq [Bertot 1999; Amerkad et al]
- Predicate Prover [Abrial and Cansell 2003]
- TPE / Epgy [Sommer and Nuckols 2004]
- Papuq [Chrzaszcz and Sakowicz 2007]

Input language

- Bare proof terms
- Core tactic-language
- Declarative style
- Controlled Natural Languages (CNL)
- Point-and-click graphical input

Proof of [instructions](#) [help to seizure](#)
 $\forall E F:R_linear_space, \forall f:E \rightarrow F,$
 $f_is_a_linear_map \Rightarrow Im(f)_is_a_linear_subspace$
 State 5

Exhibit 0

Context:

- $E: R_linear_space$
- $E: R_linear_space$
- $E \Rightarrow F$
- $H_0: f(0) = 0$
- $H_1: \forall u v : E, f(u+v) = f(u)+f(v)$
- $H_2: \forall u : E, \forall a : R, f(a*u) = a*f(u)$
- Goals: $\exists u : E, f(u) = 0$**

[- exhibit \(?\)](#) [- easy \(?\)](#)
[- translate the goal \(?\)](#) [- rewrite the goal using an equality \(?\)](#)
[- translate an hypothesis \(?\)](#)

[Go back](#)

Next purposes:

$u + v \in Im(f)$
 $a * u \in Im(f)$

- CtCoq / PCoq [Bertot 1999; Amerkad et al.]
- Predicate Prover [Abrial and Cansell 2003]
- TPE / Epgy [Sommer and Nuckols 2004]
- Papuq [Chrzaszcz and Sakowicz 2007]
- Coqweb [Blanc et al. 2007]

Input language

- Bare proof terms
- Core tactic-language
- Declarative style
- Controlled Natural Languages (CNL)
- Point-and-click graphical input

The screenshot shows the ProofLab interface. The main window is titled "ProofLab" and contains a "Derivation" section with "Info", "Edit", and "Options" buttons. The "Rule Preview" section on the left shows a definition for "definiendum" and a goal "a ∈ p(N)". The "Rules" section in the center lists "Logic Rules" and "Special Rules" with buttons for each. The "Goal" section on the right shows a list of goals: 1. $(\forall x \in a) \forall x \in b$, 2. $a \subseteq b$, and 3. $a \in p(N)$. The "Problem Score" is 0/1.

- CtCoq / PCoq [Bertot 1999; Amerkad et al.]
- Predicate Prover [Abrial and Cansell 2003]
- TPE / Epgy [Sommer and Nuckols 2004]
- Papuq [Chrzaszcz and Sakowicz 2007]
- Coqweb [Blanc et al. 2007]
- ProofLab [Sieg 2007]

Input language

- Bare proof terms
- Core tactic-language
- Declarative style
- Controlled Natural Languages (CNL)
- Point-and-click graphical input

Term editor

Logic and sets **Variables**

\forall	\exists	\top	\perp	\wedge	\vee	\neg
\Rightarrow	\Leftrightarrow	$=$	\neq	\in	\notin	\subset
$\not\subset$	\subseteq	$\not\subseteq$	\cup	\cup	\cap	\cap
\setminus	\emptyset	\times	\mathcal{P}	\circ	$\langle \rangle$	$\{ \}$

$\neg \exists A \forall B$ (\Leftrightarrow)

Cancel OK

- CtCoq / PCoq [Bertot 1999; Amerkad et al.]
- Predicate Prover [Abrial and Cansell 2003]
- TPE / Epgy [Sommer and Nuckols 2004]
- Papuq [Chrzaszcz and Sakowicz 2007]
- Coqweb [Blanc et al. 2007]
- ProofLab [Sieg 2007]
- EasyProve [Materzok 2015]

Input language

- Bare proof terms
- Core tactic-language
- Declarative style
- Controlled Natural Languages (CNL)
- Point-and-click graphical input

Edukera

The screenshot shows the Edukera proof assistant interface. The main window displays a proof script with the following content:

```

x ∈ B
f(x) = x
(2) (4) par définition d'une fo
  >> à justifier
  >> à justifier
  >> à justifier
g ∘ f (x) = g ∘ f (y)
(x) = (y)
x = y
g est injective
x ... (10) par définition d'une fonction injective
  
```

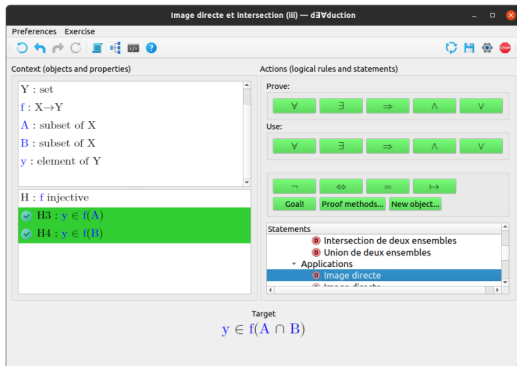
A dialog box titled "Déduction à partir de (2)" is open, showing a goal $f(x) = y$ and a tactic "par définition d'une fo". The dialog also includes a "référence" section with "à justifier" and a "déclaration" section with "par définition d'une fo".

- CtCoq / PCoq [Bertot 1999; Amerkad et al.]
- Predicate Prover [Abrial and Cansell 2003]
- TPE / Epgy [Sommer and Nuckols 2004]
- Papuq [Chrzaszcz and Sakowicz 2007]
- Coqweb [Blanc et al. 2007]
- ProofLab [Sieg 2007]
- EasyProve [Materzok 2015]
- Edukera [Rognier and Duhamel 2016]

Edukera

Input language

- Bare proof terms
- Core tactic-language
- Declarative style
- Controlled Natural Languages (CNL)
- Point-and-click graphical input



- CtCoq / PCoq [Bertot 1999; Amerkad et al.]
- Predicate Prover [Abrial and Cansell 2003]
- TPE / Epgy [Sommer and Nuckols 2004]
- Papuq [Chrzaszcz and Sakowicz 2007]
- Coqweb [Blanc et al. 2007]
- ProofLab [Sieg 2007]
- EasyProve [Materzok 2015]
- Edukera [Rognier and Duhamel 2016]
- Deaduction (Leroux) [Kerjean et al. 2022]
- ...

Feedback

- Basic feedback : proof check, proof navigation, proof state

```

def converges_to (u:sequence) (l : ℝ) : Prop :=
  ∀ ε:ℝ, ε > 0 → ∃ n0 : ℕ , ∀ n:N, n ≥ n0 → |u n - l| ≤ ε

theorem uniqueness_of_limits (u : ℕ → ℝ) (l l' : ℝ)
  (h : converges_to u l) (h' : converges_to u l') :          l = l' :=
  have h0 : ∀ ε > 0, |l - l'| ≤ ε :=
  λ (ε:ℝ) (hε : ε > 0) ↦
    have hε2 : ε/2 > 0 := by linarith
    Exists.elim (h (ε/2) hε2)
  (
    sorry
  )

```

Proof state

```

u : ℕ → ℝ
l l' : ℝ
h : converges_to u l
h' : converges_to u l'
ε : ℝ
hε : ε > 0
hε2 : ε / 2 > 0
⊢ ∀ (a : ℕ), (∀ n ≥ a, |u n - l| ≤ ε / 2) → |l - l'| ≤ ε

```

Feedback

- Basic feedback : proof check, proof navigation, proof state
- Proof planning : hints, analysis of granularity

-
- Hints, proof planning, proof sketches
[F. Wiedijk 2003; Huang 1999; Edward William Ayers 2021]
 - Proof planning and adaptation of proof step granularity
[Schiller, Dietrich, and Benz Müller 2008]

Feedback

- Basic feedback : proof check, proof navigation, proof state
- Proof planning : hints, analysis of granularity
- Error diagnosis

- Models simulating student misconceptions :

Brown and Burton 1978; Farrell, Anderson, and Reiser 1984; Zinn 2006

- Carl : anti-ATP (Diproche) : introduction of false rules

[Carl 2020]

- logical fallacies (ex deduce $\exists y, \forall x, \phi$ from $\forall x, \exists y, \phi$)
- false analogies (ex : assume $(a + b)^2 = a^2 + b^2$ by false analogy with $(a + b) \cdot 2 = a \cdot 2 + b \cdot 2$)

Feedback

- Basic feedback : proof check, proof navigation, proof state
- Proof planning : hints, analysis of granularity
- Error diagnosis
- Output in NL + Formatting (LaTeX...) + selectable level of detail

- Proverb [Huang 1999]
- Theorema[Buchberger et al. 2016]
- Ganesalingam and Gowers experience
[Ganesalingam and Gowers 2013]
- [Massot 2023]
- WTT [Nederpelt 2002]

Theorem (continuous_of_dense). Let X be a topological space and let Y be a regular topological space. Let A be a dense subset of X . Let $f : X \rightarrow Y$ be a function. Assume that for all elements x of X , f is continuous at x within A . Then f is continuous.

Proof. \circ \oplus Let x be an element of X . \circ One can see it suffices to prove that for all closed neighborhoods V' of $f(x)$, there exists a neighborhood U of x such that $f[U] \subseteq V'$. \circ Let V' be a closed neighborhood of $f(x)$. \circ \oplus We obtain an open neighborhood V of x such that $f[V \cap A] \subseteq V'$. \circ We will show that V is suitable by proving that V is a neighborhood of x and $f[V] \subseteq V'$. By assumption, V is a neighborhood of x . \circ Let z be an element of V . \circ

\circ Claim: $f(z_1)$ tends to $f(z)$ as z_1 tends to z within $V \cap A$.

- By definition it suffices (1) to prove that $z \in \overline{V \cap A}$ and (2) to prove that for all neighborhoods V'' of $f(z)$, there exists a neighborhood U of z such that $f[U \cap (V \cap A)] \subseteq V''$.

expand all Collapse all that $z \in \overline{V \cap A}$.

Current proof state:

X is a topological space

Y is a regular topological space

A is a dense subset of X

$f : X \rightarrow Y$

For all elements x' of X , f is

within A

x is an element of X

V' is a closed neighborhood of

V is an open neighborhood of

$f[V \cap A] \subseteq V'$

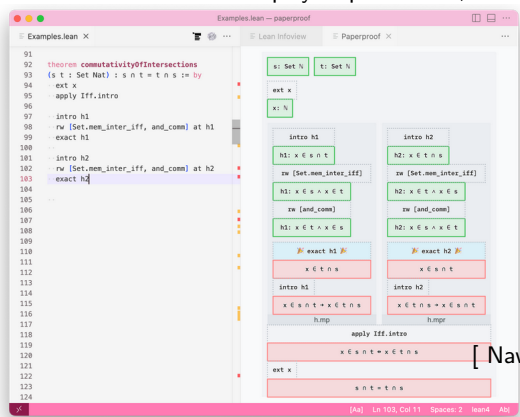
z is an element of V

Goal: $f(z_1)$ tends to $f(z)$ as

z_1 tends to z within $V \cap A$

Feedback

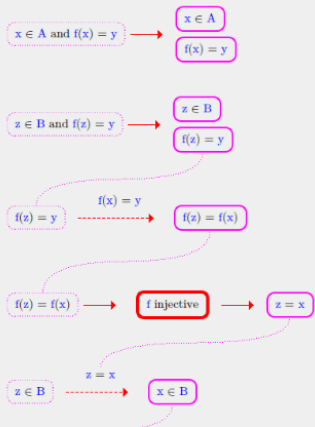
- Basic feedback : proof check, proof navigation, proof state
- Proof planning : hints, analysis of granularity
- Error diagnosis
- Output in NL + Formatting (LaTeX...) + selectable level of detail
- Visual feedback : display of proof trees, underlying objects



- Proof Visualisation
 - Paperproof [Karunus and Kovsharov 2024]
 - Deaduction [Kerjean et al. 2022]
- Underlying objects (geometry diagrams, venn diagrams, ...)
 - ProofWidgets [Nawrocki, Edward W. Ayers, and Ebner 2023]

Feedback

- Basic feedback : proof check, proof navigation, proof state
- Proof planning : hints, analysis of granularity
- Error diagnosis
- Output in NL + Formatting (LaTeX...) + selectable level of detail
- Visual feedback : display of proof trees, underlying objects



- Proof Visualisation
 - Paperproof [Karunus and Kovsharov 2024]
 - Deadduction [Kerjean et al. 2022]
- Underlying objects (geometry diagrams, venn diagrams, ...)
 - ProofWidgets [Nawrocki, Edward W. Ayers, and Ebner 2023]

Feedback

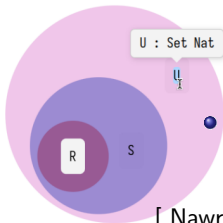
- Basic feedback : proof check, proof navigation, proof state
- Proof planning : hints, analysis of granularity
- Error diagnosis
- Output in NL + Formatting (LaTeX...) + selectable level of detail
- Visual feedback : display of proof trees, underlying objects

```
example {R S T U : Set Nat} :
  S ⊆ U → T ⊆ U → R ⊆ S → R ⊆ U := by
  withVennDisplay
  intro h1 _ h3
  exact fun n h => h ▷ h3 ▷ h1
```

▼ HTML Display

▼ Tactic state

```
R S T U : Set Nat
h1 : S ⊆ U
a f : T ⊆ U
h3 : R ⊆ S
⊢ R ⊆ U
```



• Proof Visualisation

- Paperproof

[Karunus and Kovsharov 2024]

- Deduction

[Kerjean et al. 2022]

• Underlying objects

(geometry diagrams, venn diagrams, ...)

- ProofWidgets

[Nawrocki, Edward W. Ayers, and Ebner 2023]

Underlying theory

Fundations

Sentential Logic, FOL, HOL, Dependent Type theory... :
consequences on teaching

- Naive Type theory in Papuq [Kozubek and Urzyczyn 2008; Chrzaszcz and Sakowicz 2007]

Covered topics

logics, sets, relations, functions, sequences, analysis, polynomials,
...

- Comprehensiveness and adaptation of the libraries

Handling and formalising partial functions and undefined terms

- Proof Obligations in TPE/Epgy [Sommer and Nuckols 2004]
- Partial setoids in Matita [Coen and Zoli 2007]

Teaching environment related features

Type of installation

- Standalone, locally installed
- Plugin (VsCode), locally installed
- Web app, (running on client browser or on server)

Teaching software environment

- Exercise Database (Edukera, ProofLab, Deadduction,)
- Grading feature (edukera)
- Student progress tracking, Moodle access (edukera, trylogic)
- Teacher mode (axiom and rule set customization, hint level parametrization) (waterproof, coqweb, edukera)
- Data collection (proofBuddy Karsten et al. [2023](#))
- Mixed Document feature (jsCoq, Waterproof)

Conclusion

Use of proof assistants for teaching :

Opportunity for the student to receive quick, frequent, individualized feedback

Conclusion

Use of proof assistants for teaching :

Opportunity for the student to receive quick, frequent, individualized feedback

Logic, CS : Widespread

Undergraduate maths :
remained in the community of
researchers who developed tools

Conclusion

Use of proof assistants for teaching :

Opportunity for the student to receive quick, frequent, individualized feedback

Logic, CS : Widespread

Undergraduate maths :

remained in the community of
researchers who developed tools

Why ?

Conclusion

Use of proof assistants for teaching :

Opportunity for the student to receive quick, frequent, individualized feedback

Logic, CS : Widespread

Undergraduate maths :

remained in the community of
researchers who developed tools

Why ?

Still Theoretical issues

- lack of suitable libraries adapted to the classroom
- formalization : partial functions

Conclusion

Use of proof assistants for teaching :

Opportunity for the student to receive quick, frequent, individualized feedback

Logic, CS : Widespread

Undergraduate maths :

remained in the community of
researchers who developed tools

Why ?

Still Theoretical issues

- lack of suitable libraries adapted to the classroom
- formalization : partial functions

What could help to spread ?

- Develop shared reference resources (like Software Foundations)
- Favor the development of easy to use tools, with no installation
- Involve didacticians to evaluate tools in the classroom
- Collaborate with teachers
 - raising awareness of the role of logic in proof
 - training in the use of proof assistants

Questions ?

Thank you for your attention.

Bibliography I



Abrial, Jean-Raymond and Dominique Cansell (2003). “Click’n Prove: Interactive Proofs within Set Theory”. In: *Theorem Proving in Higher Order Logics*. Ed. by David Basin and Burkhart Wolff. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, pp. 1–24. ISBN: 978-3-540-45130-3. DOI: [10.1007/10930755_1](https://doi.org/10.1007/10930755_1).







Amerkad, Ahmed et al. (June 2001). “Mathematics and Proof Presentation in Pcoq”. In: *Workshop Proof Transformation and Presentation and Proof Complexities in Connection with IJCAR 2001*. Siena.



Ayers, Edward William (Sept. 2021). “A Tool for Producing Verified, Explainable Proofs”. PhD thesis. Corpus Christi College - University of Cambridge. DOI: [10.17863/CAM.81869](https://doi.org/10.17863/CAM.81869).

Bibliography II

-  Bertot, Yves (Sept. 1999). “The CtCoq System: Design and Architecture”. In: *Formal Aspects of Computing* 11.3, pp. 225–243. ISSN: 0934-5043, 1433-299X. DOI: [10.1007/s001650050049](https://doi.org/10.1007/s001650050049). (Visited on 04/09/2024).
-  – (Mar. 2015). “Semantics for Programming Languages with Coq Encodings”. Master. France.
-  Blanc, Jérémy et al. (2007). “Teaching Proofs for Freshmen with Coqweb”. In: *Proceedings PATE07*, pp. 93–107.
-  Breitner, Joachim (2016). “Visual Theorem Proving with the Incredible Proof Machine”. In: *Interactive Theorem Proving*. Ed. by Jasmin Christian Blanchette and Stephan Merz. Vol. 9807. Karlsruhe Institute of Technology: Springer International Publishing, pp. 123–139. ISBN: 978-3-319-43143-7 978-3-319-43144-4. DOI: [10.1007/978-3-319-43144-4_8](https://doi.org/10.1007/978-3-319-43144-4_8). (Visited on 04/09/2024).

Bibliography III



Brown, John Seely and Richard R. Burton (Apr. 1978).
“Diagnostic Models for Procedural Bugs in Basic Mathematical Skills*” . In: *Cognitive Science* 2.2, pp. 155–192. ISSN: 0364-0213, 1551-6709. DOI: [10.1207/s15516709cog0202_4](https://doi.org/10.1207/s15516709cog0202_4).
(Visited on 04/15/2024).



Bruijn, de, N.G. (1968). *Automath : A Language for Mathematics*.
EUT Report. WSK, Dept. of Mathematics and Computing
Science. Technische Hogeschool Eindhoven.






Bruijn, N. G. de (Nov. 1969). “Verification des textes
mathematiques par un ordinateur” . In: Lille.



Buchberger, Bruno et al. (2016). “Theorema 2.0:
Computer-Assisted Natural-Style Mathematics” . In: *Journal of
Formalized Reasoning* 9.1, pp. 149–185.

Bibliography IV

-  Carl, Merlin (July 2020). *Automatized Evaluation of Formalization Exercises in Mathematics*. arXiv: 2006.01800 [cs, math]. (Visited on 04/02/2024).
-  Carl, Merlin, Hinrich Lorenzen, and Michael Schmitz (Feb. 2022). “Natural Language Proof Checking in Introduction to Proof Classes – First Experiences with Diproche”. In: *Electronic Proceedings in Theoretical Computer Science* 354, pp. 59–70. ISSN: 2075-2180. DOI: 10.4204/EPTCS.354.5. (Visited on 05/25/2023).
-  Chrzaszcz, Jacek and Jakub Sakowicz (2007). “Papuq: A Coq Assistant ?” In: *PATE’07*.

Bibliography V

-  Coen, Claudio Sacerdoti and Enrico Zoli (2007). “A Note on Formalising Undefined Terms in Real Analysis”. In: *PATE'07 International Workshop on Proof Assistants and Types in Education - June 25th, 2007 - The 2007 Federated Conference on Rewriting, Deduction and Programming*.
-  Delahaye, David, Mathieu Jaume, and Virgile Prevosto (Nov. 2005). “Coq, Un Outil Pour l'enseignement. Une Expérience Avec Les Étudiants Du DESS Développement de Logiciels Srs.”. In: *Technique et Science Informatiques* 24, pp. 1139–1160. DOI: [10.3166/tsi.24.1139-1160](https://doi.org/10.3166/tsi.24.1139-1160).
-  Farrell, Robert G, John R Anderson, and Brian J Reiser (1984). “1984 - An Interactive Computer-Based Tutor for LISP”. In:

Bibliography VI



From, Asta Halkjær, Jørgen Villadsen, and Patrick Blackburn (Oct. 2020). “Isabelle/HOL as a Meta-Language for Teaching Logic”. In: *Electronic Proceedings in Theoretical Computer Science* 328, pp. 18–34. ISSN: 2075-2180. DOI: [10.4204/EPTCS.328.2](https://doi.org/10.4204/EPTCS.328.2). (Visited on 04/16/2024).



Ganesalingam, M. and W. T. Gowers (Sept. 2013). *A Fully Automatic Problem Solver with Human-Style Output*. arXiv: [1309.4501 \[cs\]](https://arxiv.org/abs/1309.4501). (Visited on 03/30/2023).






Hanna, Gila and Xiaoheng Yan (Nov. 2021). “Opening a Discussion on Teaching Proof with Automated Theorem Provers”. In: *For the Learning of Mathematics*.







Huang, Xiaorong (1999). “Human Oriented Proof Presentation: A Reconstructive Approach”.






Bibliography VII

-  Karsten, Nadine et al. (Aug. 2023). “ProofBuddy: A Proof Assistant for Learning and Monitoring”. In: *Electronic Proceedings in Theoretical Computer Science* 382, pp. 1–21. ISSN: 2075-2180. DOI: [10.4204/EPTCS.382.1](https://doi.org/10.4204/EPTCS.382.1). (Visited on 02/26/2024).
-  Karunus, Evgenia and Anton Kovsharov (2024). *Lean Paperproof*. <https://github.com/Paper-Proof/paperproof>.
-  Kerjean, Marie et al. (Aug. 2022). “Utilisation Des Assistants de Preuves Pour l’enseignement En L1 - Retours d’expériences”. In: *Gazette SMF*.






Bibliography VIII

-  Kozubek, Agnieszka and Paweł Urzyczyn (2008). “In the Search of a Naive Type Theory”. In: *Types for Proofs and Programs*. Ed. by Marino Miculan, Ivan Scagnetto, and Furio Honsell. Vol. 4941. PATE'07. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 110–124. ISBN: 978-3-540-68084-0 978-3-540-68103-8. DOI: [10.1007/978-3-540-68103-8_8](https://doi.org/10.1007/978-3-540-68103-8_8). (Visited on 03/12/2024).
-  Kühlwein, Daniel et al. (2009). “The Naproche System”. In:  Lyaletski, Alexander and Konstantin Verchinine (May 2010). *Evidence Algorithm and System for Automated Deduction: A Retrospective View*. arXiv: [1005.4447 \[cs\]](https://arxiv.org/abs/1005.4447). (Visited on 04/07/2024).
-  Macbeth, Heather (2023). *The Mechanics of Proof*. <https://hrmacbeth.github.io/math2001/index.html>.

Bibliography IX

-  Massot, Patrick (2021). *Verbose Lean 4*.
<https://github.com/PatrickMassot/verbose-lean4>.
-  – (2023). *Formal Mathematics for Mathematicians and Mathematics Students*. Institute for Pure & Applied Mathematics (IPAM).
-  Materzok, Marek (2015). “Easyprove: A Tool for Teaching Precise Reasoning”. In:
-  Matuszewski, Roman and Piotr Rudnicki (Mar. 2005). “MIZAR: The First 30 Years”. In: *Mechanized Mathematics and Its Applications 4*, pp. 3–24.
-  Nawrocki, Wojciech, Edward W. Ayers, and Gabriel Ebner (2023). “An Extensible User Interface for Lean 4”. In: 20 pages, 1531949 bytes. ISSN: 1868-8969. DOI: [10.4230/LIPICS.ITP.2023.24](https://doi.org/10.4230/LIPICS.ITP.2023.24). (Visited on 02/27/2024).

Bibliography X

-  Nederpelt, Rob (May 2002). “Weak Type Theory: A Formal Language for Mathematics”. In:
-  Pierce, Benjamin C. (July 2022). *Software Foundations, 15 Years On*.
-  Portegies, Jim et al. (2022). *Waterproof: Educational Software for Learning How to Write Mathematical Proofs*.
-  Retel, Krzysztof and Anna Zalewska (Jan. 2005). “Mizar as a Tool for Teaching Mathematics”. In: *Mechanized Mathematics and Its Applications 4 (1)*, pp. 35–42.
-  Richard, P. R. and J. M. Fortuny (Jan. 2007). “Amélioration des compétences argumentatives à l’aide d’un système tutoriel en classe de mathématique au secondaire”. In: *Annales de didactique et de sciences cognitives - Revue internationale de didactique des mathématiques* 12, pp. 83–216. ISSN: 0987-7576.

Bibliography XI



Rognier, Benoit and Guillaume Duhamel (Jan. 2016).

“Présentation de La Plateforme Edukera”. In: *Vingt-Septièmes Journées Francophones Des Langages Applicatifs (JFLA 2016)*. Ed. by Julien Signoles. Saint-Malo, France.







Schiller, Marvin, Dominik Dietrich, and Christoph Benzmüller (2008). “Proof Step Analysis for Proof Tutoring - a Learning Approach to Granularity”. In: *Teaching Mathematics and Computer Science 6.2*, pp. 325–343. ISSN: 15897389. DOI: [10.5485/TMCS.2008.0183](https://doi.org/10.5485/TMCS.2008.0183). (Visited on 03/25/2024).







Sieg, W. (July 2007). “The AProS Project: Strategic Thinking & Computational Logic”. In: *Logic Journal of IGPL 15.4*, pp. 359–368. ISSN: 1367-0751, 1368-9894. DOI: [10.1093/jigpal/jzm026](https://doi.org/10.1093/jigpal/jzm026). (Visited on 03/27/2024).



Bibliography XII

-  Sommer, Richard and Gregory Nuckols (2004). “A Proof Environment for Teaching Mathematics”. In: *Journal of Automated Reasoning* 32, pp. 227–258.
-  The Cocorico Coq wiki (2021). *Coq in the Classroom*.
<http://coq.inria.fr/cocorico/CoqInTheClassroom>.
-  Thoma, Athina and Paola Iannone (Apr. 2022). “Learning about Proof with the Theorem Prover LEAN: The Abundant Numbers Task”. In: *International Journal of Research in Undergraduate Mathematics Education* 8. DOI:
[10.1007/s40753-021-00140-1](https://doi.org/10.1007/s40753-021-00140-1).
-  Trybulec, Andrzej (1983). “On a System of Computer-Aided Instruction of Logic”. In: *Bulletin of the Section of Logic* 12.4, pp. 214–218.

Bibliography XIII

-  Wagner, Marc, Serge Autexier, and Christoph Benz Müller (May 2007). “PlatΩ: A Mediator between Text-Editors and Proof Assistance Systems”. In: *Electronic Notes in Theoretical Computer Science* 174.2, pp. 87–107. ISSN: 15710661. DOI: [10.1016/j.entcs.2006.09.024](https://doi.org/10.1016/j.entcs.2006.09.024). (Visited on 04/08/2024).
-  Webber C., N. et al. (2001). “The Baghera Project: A Multi-Agent Architecture for Human Learning”. In: *Proceedings of the Workshop Multi-Agent Architectures for Distributed Learning Environments*, pp. 12–17.
-  Wenzel, Makarius (Jan. 2007). “Isabelle/Isar — a Generic Framework for Human-Readable Proof Documents”. In:
-  Wiedijk, Cezary Kaliszyk Freek (2007). “Teaching Logic Using a State-of-the-Art Proof Assistant”. In: *PATE’07*.

Bibliography XIV

-  Wiedijk, Freek (2003). “Formal Proof Sketches”. In: *International Workshop on Types for Proofs and Programs*. Springer, pp. 378–393.
-  Zinn, Claus (2006). “Supporting Tutorial Feedback to Student Help Requests and Errors in Symbolic Differentiation”. In: *Intelligent Tutoring Systems, 8th International Conference, ITS 2006, Jhongli, Taiwan, June 26-30, 2006, Proceedings*. Ed. by Mitsuru Ikeda, Kevin D. Ashley, and Tak-Wai Chan. Vol. 4053. Lecture Notes in Computer Science. Springer, pp. 349–359. ISBN: 3-540-35159-0. DOI: [10.1007/11774303_35](https://doi.org/10.1007/11774303_35).